## **Intro**

To complete this exercise you are provided with some resource files.

Please go over the resources before continuing reading

Please go over All the steps before you begin implementing

General Notes:

* All 3rd party API Calls are via their respective official Rest API
* All Config\Files files should be stored in JSON format

You will be tested on:

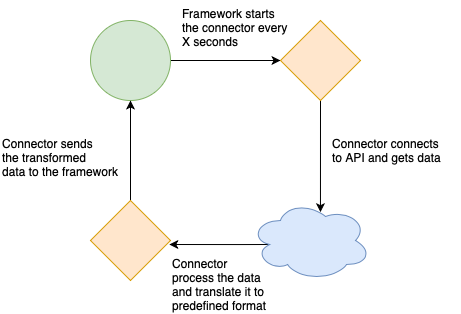
* Code cleanliness & readability
* Correct layers design & design patterns
* Error handling
* The ability to understand what is asked from you
* Production code best practices (ie: logging, input validations, etc.)

Good luck!

## **The Task - Create a Connector**

A Connector is a script with some input parameters, that is being run periodically and scheduled by some management framework.

The connectors’ goal is to connect to some 3rd party product, fetch and process data from it using its API, and deliver that data back to the framework (in the data format that the framework expects).



In this exercise, the communication with the framework will done via the SubProcessInputOutputHandler class.

A Stub is available for you in theattached resources.

* Use SubProcessInputOutputHandler.connector\_params at the start of the script to get the connector inputed parameters.
* Use SubProcessInputOutputHandler.end at the end of the script to return the output to the framework.

The framework runs the connector periodically every X seconds (ie: every 10 seconds).

Please note: no need to write the framework. Only the connector

In each iteration, the connector’s main method is called.

You case use the ***VirusTotalConnector.py*** stub from the Resources folder

### **Connector - Virus Total**

The objective of this connector is, in each iteration, read a list of entities (domains) from a file, and for each entity scan it using the[**VirusTotal**](https://developers.virustotal.com/v3.0/reference#overview)API (https://www.virustotal.com/api/v3/domains/), and return data about suspicious entities, as part of the ConnectorResult output.

* Source Files:
  + All sources files are stored in a single folder, its path is given as a parameter to the connector
  + In each iteration, the connector should process a single file from the source folder
  + Each file will contain a list of entities (domains - ie: [msn.com](http://msn.com))
  + Choose any format you want for the files (ie: json, CSV, free text)
  + At the end of each iteration, add “.done” suffix to the file name to mark it as done
  + From each file, read only a limited number of entities, the limit should be provided via a paremeter
* For each hostname, query the VirusTotal API and return result as part of the ConnectorResult
  + Result format is a dictionary of string:any, when the key is the entity (hostname) and the value is:
    - if an entity is suspicious (define suspiciousness based on “reputation” field) : return it with relevant data (of your choosing)
    - If an entity is not suspicious: return it with “Not Suspicious”
* For this step, you can implement the SubProcessInputOutputHandler to input\output from hardcoded files, and just run the *VirusTotalConncetor.py* manually**.**